Applicants: JOURDAN et al

Serial No. 09/708,722

Response to Office Action mailed May 20, 2004

REMARKS

The application contains claims 1-19. In view of the following remarks, Applicants

respectfully request allowance of the application.

INTERVIEW SUMMARY

At the outset, Applicants thank Examiner O'Brien for the courtesy of the July 14, 204,

interview with the undersigned. At that interview, Applicants compared the claims to the cited

art. The substance of these arguments is repeated below. Examiner O'Brien indicated the

arguments would be considered carefully following submission of a formal Response.

CLAIM OBJECTIONS AND §112 REJECTIONS

The foregoing amendments are provided to overcome the claim objections and §112

rejections made in the Office Action. By and large, the amendments adopt suggestions made in

the Office Action.

THE CLAIMS DEFINE OVER THE PRIOR ART

All claims stand rejected as obvious over Patel, Johnson (U.S.P. 5,924,092) and, for

some claims, Peleq. Patel and Peleq were addressed at length in Applicants' prior Response.

As noted therein and acknowledged in the most recent Office Action, Patel and Peleg

collectively fail to teach or suggest storing instructions of an instruction segment in reverse

program order. The Office Action now argues that Johnson provides adequate suggestion to

teach this subject matter. Applicants respectfully disagree.

Johnson's disclosure relates to a technique for storage of array elements. He explains

that array elements are to be sorted according to a relative frequency by which the array

elements probably will be modified. Those elements that will be modified frequently should be

stored toward the end of an array. See, Johnson, Col. 2:5-11 and 2:56-61. For some reason,

largely unexplained, Johnson believes that within a memory page (typically, a 4 KB section of

memory), data at the beginning of a logical page of memory is modified more frequently than

Applicants: JOURDAN et al Serial No. 09/708,722

Response to Office Action mailed May 20, 2004

data toward the end of the logical page. Col. 3:54-65. So, in keeping with the general teaching, data elements within a logical are stored in reverse order to place more frequently modified elements toward the end. When those elements are modified and re-stored, changes in element size do not require much shifting among the other elements as had been done in the systems discussed with respect to <u>Johnson</u>'s FIG. 1. <u>Johnson</u>, even when considered in combination with <u>Patel</u> and <u>Peleg</u>, does not render the pending claims obvious.

<u>Johnson</u>'s disclosure does not provide an adequate teaching or suggestion to modify any of <u>Patel</u>'s features to arrive at the invention. <u>Johnson</u> nowhere refers to program instructions. <u>Johnson</u> teaches to sort elements of a data array to place the elements most likely to be modified toward the end of the array. Data arrays commonly store data variables, not program instructions. <u>Johnson</u>'s techniques allegedly limit the amount of rearrangement that becomes necessary if an element were changed to a new size and re-stored in the array. Such teachings, however, have no application toward the <u>Patel</u> system. <u>Patel</u>'s traces store program instructions. Program instructions are not modified. They do not get longer or shorter. They are simply instructions. Accordingly, the data arrangement problems noted by <u>Johnson</u> do not occur in <u>Patel</u>'s trace-based system.

The Office Action places undue emphasis on <u>Johnson</u>'s statement that elements from a page of memory can be stored in reverse order. First, <u>Johnson</u> is referring to a 4 KB page of memory. It is unclear what bearing this has to a system such as <u>Patel</u>'s where each trace can store at most 16 instructions. Second, <u>Johnson</u>'s reverse order idea is based on an expectation that page elements at the 'top' of a page are most frequently modified. As noted above, program instructions within a trace are not modified.¹ When read in context, it becomes clear that Johnson's "reverse order" statement has no relevance to the claimed subject matter.

¹Although Patel does not fully describe his build process, it is believed that traces are not modified after they are stored in a trace cache. Instead, an old trace can be replaced by a similar new trace, which is built through independent operation. Alternatively, multiple copies of similar traces could be stored in different locations of the trace cache. Patel describes this somewhat in § 5 & FIG. 8.

Applicants: JOURDAN et al Serial No. 09/708,722

Response to Office Action mailed May 20, 2004

Finally, Applicants note the references do not provide adequate motivation to consider teachings of these three references in combination. <u>Patel</u> and <u>Peleg</u> are directed to trace caches within microprocessors, which typically store at most 16 instructions at a time. <u>Johnson</u> is directed to management of data arrays in system memory. There is no suggestion anywhere to adopt any teaching from Johnson into any other component within a computer system.

All claims recite that instruction segments store their instructions in reverse program order. The cited art does not teach or suggest this subject matter. Accordingly, the outstanding rejections should be withdrawn.

Applicants respectfully request allowance of the application.

Respectfully submitted,

Date: 8/20/09

Robert L. Hails, Jr.

Registration No. 39,702

(Attorney for Intel Corporation)

KENYON & KENYON 1500 K Street, N.W. Washington, D.C. 20005

Ph.: (202) 220-4200 Fax.: (202) 220-4201